

Package ‘psc’

November 20, 2024

Type Package

Title Personalised Synthetic Controls

Version 1.0.0

Maintainer Richard Jackson <richJ23@liverpool.ac.uk>

Description Allows the comparison of data cohorts (DC) against a Counter Factual Model (CFM) and measures the difference in terms of an efficacy parameter. Allows the application of Personalised Synthetic Controls.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

URL <https://github.com/richjjackson/psc/>

BugReports <https://github.com/richJJackson/psc/issues>

Depends R (>= 3.5.0)

Imports mvtnorm, survival, enrichwith, stats, flexsurv

Suggests knitr, rmarkdown, devtools

VignetteBuilder knitr

NeedsCompilation no

Author Richard Jackson [cre, aut] (<<https://orcid.org/0000-0002-7814-5088>>)

Repository CRAN

Date/Publication 2024-11-20 10:40:02 UTC

Contents

acc	2
bin.mod	3
coef.psc	4
cont.mod	4
count.mod	5

data	6
dataComb	7
dataComb.flexsurvreg	7
dataComb.glm	8
initParm	9
initParm.flexsurvreg	9
initParm.glm	10
lik.flexsurvreg	11
lik.flexsurvreg.mtc	11
lik.glm	12
lik.glm.mtc	12
linPred	13
modelExtract	13
modelExtract.flexsurvreg	14
modelExtract.glm	15
modp	15
plot.psc	16
print.psc	16
psc.object	17
pscEst	17
pscEst.flexsurvreg	18
pscEst.glm	19
pscfite	20
pscSumm	21
psc_ite	22
summary.psc	22
surv.mod	23
surv_fpm	24
Index	25

acc

acc

Description

acc

Usage

acc(old, new)

Arguments

old	a numeric value
new	a numeric value

Details

A function for the evaluation of two likelihoods as part of the MCMC procedure

Value

returns the an evaluation of $\text{old/new} > U$ where U is a draw from the uniform distribution

bin.mod

Example model for a survival outcome

Description

A model of class 'glm'

Usage

bin.mod

Format

A model of class 'flezsurgreg':

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastasis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count exapmple outcome for count data

trt exapmple identifier for mulitple treatment comparisons

aet Aetiology

Source

simulated

coef.psc	Returns the coefficient estimate of a psc object.
----------	---

Description

Returns the coefficient estimate of a psc object.

Usage

```
## S3 method for class 'psc'
coef(object, ..., level = 0.05)
```

Arguments

object	a 'psc' object
...	not used
level	the level at which credibility intervals are assessed, defaults to 0.05

Value

The summary of the posterior distribution for the efficacy parameter in terms of the median and 95

cont.mod	cont.mod
----------	----------

Description

A model of class 'glm'

Usage

```
cont.mod
```

Format

A model of class 'flezsurgreg':

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastasis
ageVasInv centred age nested within vasculdevar invasion
time survival time
cen censoring indicator
os survival time
count exapmple outcome for count data
trt exapmple identifier for mulitple treatment comparisons
aet Aetiology

Source

simulated

count.mod	<i>Example model for a survival outcome</i>
-----------	---

Description

A model of class 'glm'

Usage

count.mod

Format

A model of class 'flezsurgreg':

gamma cumulative baseline hazard parameters
vi vascular invasion
age60 patient age (centred at 60)
ecog ECOG performance Status
logafp AFP - log scale
alb albumin
logcreat Creatinine - log scale
allmets metastasis
ageVasInv centred age nested within vascular invasion
time survival time
cen censoring indicator
os survival time
count exapmple outcome for count data
trt exapmple identifier for mulitple treatment comparisons
aet Aetiology

Source

simulated

data

Example Dataset of patients with aHCC receiving Lenvetanib

Description

A dataset containing 100 simulated patients.

Usage

data

Format

A model of class 'flezsurgreg':

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastasis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count example outcome for count data

trt example identifier for multiple treatment comparisons

aet Aetiology

Source

simulated

dataComb	<i>A generic function for cleaning data ready for analysis</i>
----------	--

Description

A generic function for cleaning data ready for analysis

Usage

```
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

CFM	a model object supplied to pscfit
DC	a dataset including covariates to match the CFM
id	to specify which observations in the data cohort should be evaluated. Defaults to 'NULL' i.e all observations
trt	used to specify multiple treatment effects. Defaults to NULL

Value

datComb returns a list containing objects which detail the components of both the Counter Factual Model (CFM) and the Data Cohort (DC) the required exported components of the model and a cleaned data cohort.

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod,data)
```

dataComb.flexsurvreg	<i>Fucntion for cleaning the data of a model with class 'flexsurvreg'</i>
----------------------	---

Description

Fucntion for cleaning the data of a model with class 'flexsurvreg'

Usage

```
## S3 method for class 'flexsurvreg'
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

CFM	a model object supplied to pscfit
DC	a dataset including covariates to match the CFM
id	a vector specifying whether a subset of the dataset should be selected. Defaults to 'NULL' e.g. all data points included
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Value

a list containing objects which specify the required exported components of the model and a cleaned data cohort.

dataComb.glm	<i>Fucntion for cleaning the data of a model with class 'flexsurvreg'</i>
--------------	---

Description

Fucntion for cleaning the data of a model with class 'flexsurvreg'

Usage

```
## S3 method for class 'glm'
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

CFM	a model object supplied to pscfit
DC	a dataset including covariates to match the CFM
id	to specify which observations in the data cohort should be evaluated. Defaults to 'NULL' i.e all observations
trt	used to specify multiple treatment effects. Defaults to NULL

Value

a list containing objects which specify the required exported components of the model and a cleaned data cohort.

initParm	<i>Fuction for estimating initial parameter values 'flexsurvreg'</i>
----------	--

Description

Fuction for estimating initial parameter values 'flexsurvreg'

Usage

```
initParm(CFM, DC_clean, trt)
```

Arguments

CFM	A counter-factual model
DC_clean	a cleaned dataset object obtained using dataComb.flexsurvreg
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the likelihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

initParm.flexsurvreg	<i>Fuction for estimating initial parameter values 'flexsurvreg'</i>
----------------------	--

Description

Fuction for estimating initial parameter values 'flexsurvreg'

Usage

```
## S3 method for class 'flexsurvreg'
initParm(CFM, DC_clean, trt = NULL)
```

Arguments

CFM	A counter-factual model
DC_clean	a cleaned dataset object obtained using dataComb.flexsurvreg
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the likelihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

 initParm.glm

Function for estimating initial parameter values 'flexsurvreg'

Description

Function for estimating initial parameter values 'flexsurvreg'

Usage

```
## S3 method for class 'glm'
initParm(CFM, DC_clean, trt = trt)
```

Arguments

CFM	A counter-factual model
DC_clean	a cleaned dataset object obtained using dataComb.flexsurvreg
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the likelihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

lik.flexsurvreg	<i>Likelihood functio for a a psc model of class 'flexsurvreg'</i>
-----------------	--

Description

Likelihood functio for a a psc model of class 'flexsurvreg'

Usage

```
lik.flexsurvreg(beta, DC_clean)
```

Arguments

beta	a parameter to be estimate
DC_clean	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg'

lik.flexsurvreg.mtc	<i>Likelihood functio for a a psc model of class 'flexsurvreg'</i>
---------------------	--

Description

Likelihood functio for a a psc model of class 'flexsurvreg'

Usage

```
lik.flexsurvreg.mtc(beta, DC_clean)
```

Arguments

beta	a parameter to be estimate
DC_clean	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg' where mulitple treatment comparisons are

lik.glm	<i>Likelihood functio for a a psc model of class 'glm'</i>
---------	--

Description

Likelihood functio for a a psc model of class 'glm'

Usage

```
lik.glm(beta, DC_clean)
```

Arguments

beta	a parameter to be estimate
DC_clean	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg'

lik.glm.mtc	<i>Likelihood functio for a a psc model of class 'flexsurvreg'</i>
-------------	--

Description

Likelihood functio for a a psc model of class 'flexsurvreg'

Usage

```
lik.glm.mtc(beta, DC_clean)
```

Arguments

beta	a parameter to be estimate
DC_clean	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg' where mulitple treatment comparisons are

linPred	<i>Estimates the linear predictor of a psc object</i>
---------	---

Description

Estimates the linear predictor of a psc object

Usage

```
linPred(DC_clean, resp = FALSE)
```

Arguments

DC_clean	a cleaned data object created using dataComb()
resp	detailing whether the linear predictor should be returned on the natural or response level. Defaults to the natural scale (resp=F)

Details

A function which combines the data from the data cohort against the model parameters of the PSC

Value

Extracts the linear predictor from a object containing both a counterfactual model and a data cohort which is created using the dataComb() function.

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod, data)
lp <- linPred(dc)
```

modelExtract	<i>A generic function for extracting model information</i>
--------------	--

Description

A generic function for extracting model information

Usage

```
modelExtract(CFM)
```

Arguments

CFM a model of class either 'glm' or 'flexsurvreg'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

modelExtract.flexsurvreg

A generic function for extracting model information

Description

A generic function for extracting model information

Usage

```
## S3 method for class 'flexsurvreg'  
modelExtract(CFM)
```

Arguments

CFM a model of class 'flexsurvreg'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

modelExtract.glm	<i>A generic function for extracting model information</i>
------------------	--

Description

A generic function for extracting model information

Usage

```
## S3 method for class 'glm'
modelExtract(CFM)
```

Arguments

CFM a model of class 'glm'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

modp	<i>modp</i>
------	-------------

Description

modp

Usage

```
modp(x)
```

Arguments

x a numeric vector

Details

A function which returns a version of x with negative values replaced with 0

Value

a numeric vector with negative values replaced with 0

plot.psc *Fucntion for Plotting PSC objects*

Description

Fucntion for Plotting PSC objects

Usage

```
## S3 method for class 'psc'  
plot(x, ...)
```

Arguments

x	an object of class 'psc'
...	not used

Details

making use of the generic 'plot' functions this will provide some graphical output of the fitted psc object. The form of the output will depend on the class of the initial model

Value

a plot corresponding to the psc fit

print.psc *Personalised Synthetic Controls - print*

Description

Personalised Synthetic Controls - print

Usage

```
## S3 method for class 'psc'  
print(x, ...)
```

Arguments

x	an object of class 'psc'
...	not used

Value

printing psc results

psc.object *Fitted psc object*

Description

An object returned by the `pscf` function, inheriting from class `psc` and representing a fitted personalised synthetic control model.

Usage

`psc.object`

Format

An object of class `NULL` of length 0.

Author(s)

Richard Jaskson (<richj23@liverpool.ac.uk>)

pscEst *Function for performing estimation procedures in 'pscf'*

Description

Function for performing estimation procedures in 'pscf'

Usage

`pscEst(CFM, DC_clean, nsim, start, trt)`

Arguments

<code>CFM</code>	a model object supplied to <code>pscf</code>
<code>DC_clean</code>	a cleaned dataset obtained using <code>dataComb()</code> .
<code>nsim</code>	the number of MCMC simulations to run
<code>start</code>	the starting value for
<code>trt</code>	an optional vector denoting treatment allocations where multiple treatment comparisons are being made

Details

Define the set of model parameters B to contain Γ which summarize the parameters of the CFM. Prior distributions are defined for B using a multivariate normal distribution $\pi(B) \sim MVN(\mu, \Sigma)$ where μ is the vector of coefficient estimates from the validated model and Σ is the variance-covariance matrix. This information is taken directly from the outputs of the parametric model and no further elicitation is required. The prior distribution for the efficacy parameter ($\pi(\beta)$) is set as an uninformative $N(0, 1000)$.

Ultimately the aim is to estimate the posterior distribution for β conditional on the distribution of B and the observed data. A full form for the posterior distribution is then given as

$$P(\beta|B, D) \propto L(D|B, \beta)\pi(B)\pi(\beta)$$

Please see 'pscfitt' for more details on likelihood formation.

For each iteration of the MCMC procedure, the following algorithm is performed

1. Set an indicator $s=1$, and define an initial state based on prior hyperparameters for $\pi(B)$ and $\pi(\beta)$ such that $b_s = \mu$ and $\tau_s = 0$
2. Update $s = s + 1$ and draw model parameters b_s from $\pi(B)$ and draw a proposal estimate of β from some target distribution
3. Estimate $\Gamma(i, S) = \nu^T x_i$ where ν is the subset of parameters from b_s which relate to the model covariates and define 2 new likelihood functions $\Theta(s, 1) = L(D|B = b_s, \beta = \tau_{(s-1)})$ & $\Theta(s, 2) = L(D|B = b_s, \beta = \tau_s)$
4. Draw a single value ψ from a Uniform (0,1) distribution and estimate the condition $\omega = \Theta(s, 1)/\Theta(s, 2)$. If $\omega > \psi$ then accept τ_s as belonging to the posterior distribution $P(\beta|B, D)$ otherwise retain $\tau_{(s-1)}$
5. Repeat steps 2 – 4 for the required number of iterations

The result of the algorithm is a posterior distribution for the log hazard ratio, β , captures the variability in B through the defined priors $\pi(\beta)$.

Value

A matrix containing the draws from the posterior distribution

pscEst.flexsurvreg *Function for estimating initial parameter values 'flexsurvreg'*

Description

Function for estimating initial parameter values 'flexsurvreg'

Usage

```
## S3 method for class 'flexsurvreg'
pscEst(CFM, DC_clean, nsim, start, trt = trt)
```

Arguments

CFM	a model object supplied to pscfit
DC_clean	a cleaned dataset obtained using dataComb().
nsim	the number of MCMC simulations to run
start	the starting value for
trt	an optional vector denoting treatment allocations where multiple treatment comparisons are being made

Details

An MCMC routine for fitting a psc model

Value

A matrix containing the draws from the posterior distribution

pscEst.glm	<i>Function for estimating initial parameter values 'glm'</i>
------------	---

Description

Function for estimating initial parameter values 'glm'

Usage

```
## S3 method for class 'glm'
pscEst(CFM, DC_clean, nsim, start, trt = trt)
```

Arguments

CFM	a model object supplied to pscfit
DC_clean	a cleaned dataset obtained using dataComb().
nsim	the number of MCMC simulations to run
start	the starting value for
trt	an optional vector denoting treatment allocations where multiple treatment comparisons are being made

Details

An MCMC routine for fitting a psc model

Value

a matrix containing the draws from the posterior distribution

pscfit

*Personalised Synthetic Controls model fit***Description**

Personalised Synthetic Controls model fit

Usage

pscfit(CFM, DC, nsim = 5000, id = NULL, trt = NULL)

Arguments

CFM	A model of type 'glm' or 'flexsurvspline'
DC	A dataset including columns to match to covariates in the model
nsim	The number of simulations for the MCMC routine
id	Numeric vector stating which patient(s) from the dataset should be included in the analysis. Defaults to all patients
trt	An optional vector denoting treatment allocations for multiple treatment comparisons. Defaults to NULL.

Details

the pscfit function compares a dataset ('DC') against a parametric model. This is done by selecting a likelihood which is identified by the type of CFM that is supplied. At present, two types of model are supported, a flexible parametric survival model of type 'flexsurvreg' and a generalised linear model of type 'glm'.

Where the CFM is of type 'flexsurvreg' the likelihood supplied is of the form:

$$L(D|\Lambda, \Gamma_i) = \prod_{i=1}^n f(t_i|\Lambda, \Gamma_i)^{c_i} S(t_i|\Lambda, \Gamma_i)^{(1-c_i)}$$

Where Λ defines the cumulative baseline hazard function, Γ is the linear predictor and t and c are the event time and indicator variables.

Where the CFM is of the type 'glm' the likelihood supplied is of the form:

$$L(x|\Gamma_i) = \prod_{i=1}^n b(x|\Gamma_i) \exp \{ \Gamma_i^T t(x) - c(\Gamma_i) \}$$

Where $b(\cdot)$, $t(\cdot)$ and $c(\cdot)$ represent the functions of the exponential family. In both cases, Γ is defined as:

$$\Gamma = \gamma x + \beta$$

Where γ are the model coefficients supplied by the CFM and β is the parameter set to measure the difference between the CFM and the DC.

Estimation is performed using a Bayesian MCMC procedure. Prior distributions for Γ (& Λ) are derived directly from the model coefficients (mean and variance covariance matrix) or the CFM. A bespoke MCMC routine is performed to estimate β . Please see '?mcmc' for more details.

For the standard example where the DC contains information from only a single treatment, trt need not be specified. Where comparisons between the CFM and multiple treatments are require, a covariate of treatmne allocations must be specified sperately (using the 'trt' option).

Value

a object of class 'psc' with attributes model.type, the cleaned Dataset and the posterior distribution of the fitted model

Attributes include

- A 'cleaned' dataset including extracted components of the CFM and the cleaned DC included in the procedure
- An object defining the class of model (and therefore the procedure applied - see above)
- A matrix containing the draws of the posterior distributions

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
surv.psc <- pscfit(surv.mod,data)
```

pscSumm

Personalised Synthetic Controls - summary

Description

Personalised Synthetic Controls - summary

Usage

```
pscSumm(DC_clean)
```

Arguments

DC_clean a cleaned dataset ontained using dataComb().

Value

psc summary results including an estimate of the linear predictor combing the data and the model, an estimate of patient level response and summary statistics of the average responses for the sythentic and observed populations

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod,data)
summ <- pscSumm(dc)
```

psc_ite

Wrapper function for individual treatment effects

Description

Wrapper function for individual treatment effects

Usage

```
psc_ite(model, data)
```

Arguments

model	a CFM
data	a data cohort

Details

A wrapper function that estimates individual treatment effects using pscfit

summary.psc

Personalised Synthetic Controls - summary

Description

Personalised Synthetic Controls - summary

Usage

```
## S3 method for class 'psc'
summary(object, ...)
```

Arguments

object	an object of class 'psc'
...	not used

Value

A summary of a psc object obtained using pscSumm and a copy of the pscfit object

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
psc.ob <- pscfit(surv.mod,data)
summary(psc.ob)
```

surv.mod

Example model for a survival outcome

Description

A model of class 'flezsurrreg'

Usage

surv.mod

Format

A model of class 'flezsurrreg':

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastasis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count exapmple outcome for count data

trt exapmple identifier for mulitple treatment comparisons

aet Aetiology

Source

simulated

`surv_fpm``modp`

Description`modp`**Usage**

```
surv_fpm(DC_clean, beta = 0, s = NULL)
```

Arguments

<code>DC_clean</code>	a cleaned dataset obtained using <code>dataComb()</code> .
<code>beta</code>	a parameter to determine if the survival probabilities should be adjusted by some (log) hazard ratio. Defaults to <code>beta=0</code> , i.e. no adjustment.
<code>s</code>	if specified will return the time at which some threshold is passed (e.g. <code>s=0.5</code> for median survival time)

Details

A function which extracts survival probabilities from a `flexsurvreg` object

Value

a list of times and associated survival probabilities

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod, data)
s_est <- surv_fpm(dc)
```


Index

* datasets

- bin.mod, 3
- cont.mod, 4
- count.mod, 5
- data, 6
- psc.object, 17
- surv.mod, 23

acc, 2

bin.mod, 3

- coef.psc, 4
- cont.mod, 4
- count.mod, 5

- data, 6
- dataComb, 7
- dataComb.flexsurvreg, 7
- dataComb.glm, 8

- initParm, 9
- initParm.flexsurvreg, 9
- initParm.glm, 10

- lik.flexsurvreg, 11
- lik.flexsurvreg.mtc, 11
- lik.glm, 12
- lik.glm.mtc, 12
- linPred, 13

- modelExtract, 13
- modelExtract.flexsurvreg, 14
- modelExtract.glm, 15
- modp, 15

- plot.psc, 16
- print.psc, 16
- psc.object, 17
- psc_ite, 22
- pscEst, 17

- pscEst.flexsurvreg, 18
- pscEst.glm, 19
- pscfit, 20
- pscSumm, 21

- summary.psc, 22
- surv.mod, 23
- surv_fpm, 24